LA-UR 97-478

Approved for public release; distribution is unlimited

Service-constrained Network Design Problems

Authors: M.V. Marathe, R. Ravi, R. Sundaram

Nordic Journal of Computing

SERVICE-CONSTRAINED NETWORK DESIGN PROBLEMS

MADHAV V. MARATHE
Los Alamos National Laboratory P.O. Box 1663
MS B265, Los Alamos NM 87545, USA
madhav@c3.lanl.gov

R. RAVI
Graduate School of Industrial Administration
Carnegie Mellon University, 5000 Forbes Avenue
Pittsburgh, PA 15213, USA
ravi@andrew.cmu.edu

RAVI SUNDARAM
Delta Global Trading L. P.
Four Cambridge Center, Cambridge MA 02142, USA
koods@delta-global.com

Abstract.
Several practical instances of network design problems often require the network to satisfy multiple constraints. In this paper, we focus on the following problem (and its variants): find a low-cost network, under one cost function, that *services* every node in the graph, under another cost function, (i.e., every node of the graph is within a prespecified distance from the network). Such problems find applications in optical network design and the efficient maintenance of distributed databases.

We utilize the framework developed in Marathe *et al.* [1995] to formulate these problems as bicriteria network design problems, and present approximation algorithms for a class of service-constrained network design problems. We also present lower bounds on the approximability of these problems that demonstrate that our performance ratios are close to best possible.

Key words: Approximation algorithms, Bicriteria problems, Spanning trees, Network design, Combinatorial algorithms.

CR Classification: G.2.2.

1. Introduction and Motivation

The problem of managing replicated copies of a data in a distributed database is an important and extensively studied problem in computer science. (See Awerbuch *et al.* [1992], Awerbuch *et al.* [1993], Lund *et al.* [1994], Dowdy and Foster [1982], Milo and Wolfson [1988], Kumar and Segev [1993] and the references therein.) As an example, consider the problem posed in Wolfson and Milo [1991] on the design of distributed databases: given a set of sites in a network we wish to select a subset of the sites at which to place

Received August 1996. Revised October 1996.

copies of the database. The major requirement is that each site should be able to access a copy of the database within a prespecified service time, and the chosen sites should be connected together as a minimum cost tree so that updates to one of the copies can be propagated to the other copies in a cost effective manner (See Fig. 1.1).

A problem of a similar nature comes up in the area of optical network design. Developments in fiber-optic networking technology have finally reached the point where it is being considered as the most promising candidate for the next generation of wide-area backbone networks (Green [1992]). The optical network is a pure data transmission medium. All the computing and processing continues to be done in the electronic world. An important issue in interfacing these two worlds – the electronic and the optic – is that of designing the optical network subject to location-theoretic constraints imposed by the electronic world. Given a set of sites in a network we wish to select a subset of the sites at which to place optoelectronic switches and routers. As before, the major requirement is that every site should be within a prespecified distance or delay from an optoelectronic access node and the chosen sites should be connected together using fiber-optic links in a minimum cost tree (See Fig. 1.2).

As a final application consider the *Traveling Cameraman Problem* that arises in automatic optical inspection of printed circuit boards (see Iwano *et al.* [1994], Hernandes *et al.* [1993] and the references therein) . In this problem, a camera is positioned over the board and can be freely moved in a plane parallel to the board. The camera is moved over the board so as to photograph parts of the board at various positions. These photographs are compared to the "master photograph" for detecting possible defects such as violation of design rules, mounting and soldering condition of the components on the board, etc. An important consideration is the time taken to perform the entire inspection sequence; which is proportional to the distance traversed by the camera as well as the number of photographs taken. Thus the goal of the problem is to design a strategy for moving the camera to cover a minimum distance with the constraint that the entire board is photographed.

All of the above stated problems can be thought of as instances of "service-constrained network design problems." Informally, service-constrained network design problems involve both a location-theoretic objective and a cost-minimization objective subject to connectivity constraints. The location-theoretic objective requires that we choose a subset of nodes at which to "locate" services such that each node is within a bounded distance from at least one chosen location. The cost-minimization objective requires that the chosen locations be connected by a network minimizing parameters such as total cost, diameter or maximum edge cost. The two objectives are measured under two (possibly) different cost functions.

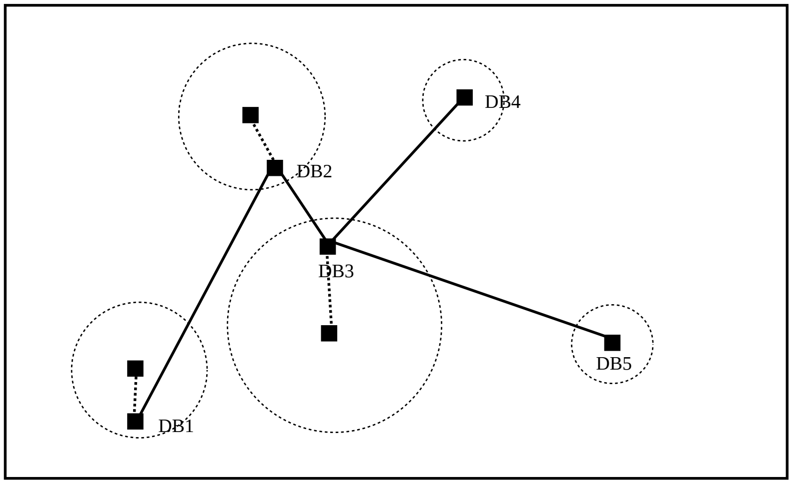

Fig. 1.1: The database copies are shown linked by a network. The circles represent the prespecified service times.

2. Problem Statement

The prototypical problem we consider in this paper is the following: We are given an undirected graph $G = (V, E)$ with two different cost functions c (modeling the service cost) and d (modeling the construction or communication cost) for each edge $e \in E$, and a bound $\mathcal{S}_v$ (on the service distance for each vertex v). The goal is to find a minimum d-cost tree such that every node v in the graph is *serviced* by some node in the tree, i.e. every node v

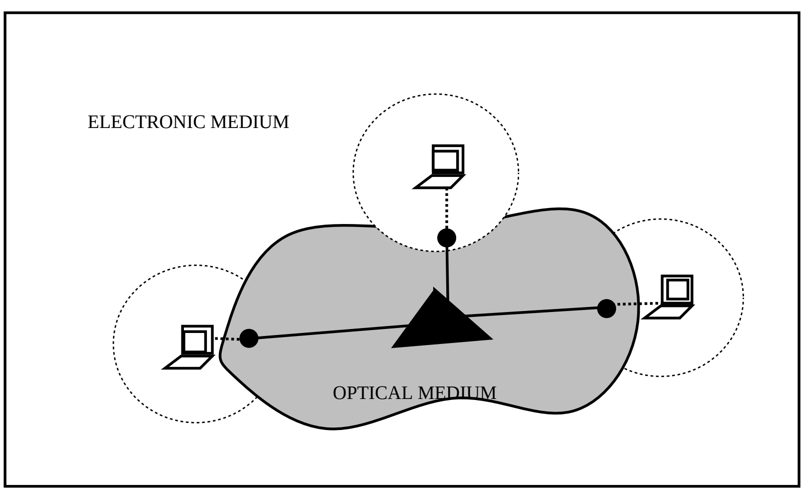

Fig. 1.2: The optoelectronic switches are shown linked by a network. The circles represent the prespecified distances.

is within distance $\mathcal{S}_v$ (under the c-costs) of some node in the tree.

We use the bicriteria framework developed in Marathe *et al.* [1995]. A generic bicriteria network design problem, (**A**, **B**, **S**), is defined by identifying two minimization objectives, – **A** and **B**, – from a set of possible objectives, and specifying a membership requirement in a class of subgraphs, – **S**. The problem specifies a budget value on the first objective, **A**, under one cost function, and seeks to find a network having minimum possible value for the second objective, **B**, under another cost function, such that this network is within the budget on the first objective **A**. The solution network must belong to the subgraph-class **S**.

The two versions of the location-theoretic or service cost objective that we consider are: (i) Non-uniform maximum service cost (denoted by **Non-uniform service cost**) and (ii) Uniform service cost (denoted by **Uniform service cost**). In the *Non-uniform service cost* version a service constraint $\mathcal{S}_{v_k}$ is specified for each vertex. The *Uniform service cost* version is a special case where $\forall v_k,\ \mathcal{S}_{v_k} = \mathcal{S}$, i.e., all vertices have the same service constraint. Thus for the problems considered in this paper $\mathbf{A} \in \{$ Non-uniform service cost, Uniform service cost $\}$. For the cost-minimization objective we focus our attention on the total cost of the network. The *Total cost* objective is the sum of the costs of all the edges in the network. We also consider the *Diameter* objective – the maximum distance between any pair of nodes in the network – and the *Bottleneck* objective – the maximum value of any edge in the network. Thus $\mathbf{B} \in \{$ **total cost, diameter, bottleneck cost** $\}$ Finally, for the problems considered here $\mathbf{S} \in \{$ **(Spanning) tree, Steiner tree, generalized Steiner tree** $\}$. For example, the problem of finding a low-cost service constrained network introduced in Figure 1.1 is the **(Non-uniform service cost, Total cost, Tree)** problem.

The organization of the rest of the paper is as follows: In Section 3, we survey related results. Section 4 provides an overview of the results in this paper. Section 5 discusses the robustness and the generality of our formulations and results. Section 6 contains the results on hardness of approximations for both the different and identical cost cases. Section 7 contains the approximation algorithms for spanning trees under different cost functions. Section 8 contains the algorithms for spanning trees generalized Steiner forests, when the cost functions are identical; Section 9 investigates the diameter and bottleneck cost objectives; Section 10 contains some concluding remarks and open problems.

3. Previous Work

Variants of the service-constrained *tour* problem have been considered by Arkin *et al.* [1994], Arkin and Hassin [1994], Current and Schilling [1989]. Current and Schilling [1989] consider the *covering salesperson problem* and present a heuristic for it without providing any performance guarantees. In this problem, nodes represent customers and the service radius represents the distance a customer is willing to travel to meet the salesperson. The goal is to find a minimum length salesperson tour so that all the (customer) nodes are strictly serviced. Arkin and Hassin [1994] considered geometric versions of the problem, where the service neighborhood (i.e., the neighborhood the customer is willing to travel) is modeled as a region in the plane. For convex neighborhoods, they present heuristics that provide constant performance guarantees. They also show how their heuristics can be extended to nonconvex regions. Arkin *et al.* [1994] considered additional geometric variations of the covering tour problem including the *lawn mower problem*, where the goal is to find a tour such that each given point within some boundary (the lawn) is within a circle of unit radius from at least one point on the tour. They provide an approximation algorithm for this problem with a constant performance guarantee. Recently Mata and Mitchell [1995] generalized and improved the results of Arkin *et al.* [1994] on geometric covering problems. Iwano *et al.* [1994] considered a geometric version of the problem motivated by applications in automatic optical inspection of printed circuit boards.

Awerbuch *et al.* [1992], Awerbuch *et al.* [1993] and Lund *et al.* [1994] consider an on-line variant of the distributed data management problem. In their model, read or write requests from various processing units arrive in an on-line fashion and an on-line algorithm needs to decide whether to replicate, move or discard copies of the database after serving each request. The goal of the on-line algorithm is to minimize the total cost of processing these requests. Due to the differences in the model considered, their results do not apply to the problems considered here.

We refer the reader to the work in Marathe *et al.* [1995] for other references on approximation algorithms for multicriteria network design.

4. Overview of Results

In this paper, we study the complexity and approximability of a number of service-constrained network design problems discussed in Section 2. Many of the problems considered in this paper, are **NP**-hard (Garey and Johnson [1979]). Given the hardness of finding optimal solutions, we concentrate on devising approximation algorithms with worst case performance guarantees. Recall that an approximation algorithm for an optimization problem Π provides a *performance guarantee* of ρ if for every instance I of Π, the solution value returned by the approximation algorithm is within a factor ρ of the optimal value for I. Define an (α, β)-approximation algorithm for an (**A**, **B**, **S**)-bicriteria problem as a polynomial-time algorithm that produces a solution in which the first objective (**A**) value, is at most α times the budget, and the second objective (**B**) value, is at most β times the minimum for any solution that is within the budget on **A**. The solution produced must belong to the subgraph-class **S**.

As mentioned before, the two objectives are measured with respect to different edge-cost functions. The (budgeted) service cost objective is measured using the c-cost function while the cost-minimization objective is measured using the d-cost function. As stated before, a node u is said to *service* node v if u is within distance $\mathcal{S}_v$ of v, under the c-cost. The *service-degree* of a node is defined to be the number of nodes it services. All our results come in two flavors: (i) Different cost functions and (ii) Identical cost functions. The *Identical cost functions* version is a special case of the *Different cost functions* case where the two cost functions are the same, i.e. $c_e = d_e, \forall e$.

We give a $(1, O(\tilde{\Delta} \ln n))$-approximation algorithm for the (**Non-uniform service cost, Total edge cost, Spanning Tree**) problem (where $\tilde{\Delta}$ is the maximum service-degree of any node in the graph). We counterbalance this by showing that even the uniform service cost version of the problem does not have an (α, β)-approximation algorithm for any $\alpha \geq 1$ and $\beta < \ln n$ unless $\mathbf{NP} \subseteq \mathbf{DTIME}(n^{\log \log n})$. When both the objectives are evaluated under the same cost function we provide a $(2(1+\epsilon), 2(1+\frac{1}{\epsilon}))$-approximation algorithm, for any $\epsilon > 0$. In the opposite direction we provide a hardness result showing that even in the restricted case where the two cost functions are the same the problem does not have an (α, β)-approximation algorithm for $\alpha < 2$ and $\beta < \ln n$ unless $\mathbf{NP} \subseteq \mathbf{DTIME}(n^{\log \log n})$. For the identical cost functions case, our method extends to generalized Steiner forest version of the problem with weaker guarantees. Finally, we show that the problems (**Non-uniform service cost, Diameter, Spanning Tree**) and (**Non-uniform service cost, Bottleneck, Spanning Tree**) are solvable in polynomial time. Again, our results extend to the Steiner tree variants of the problems.

5. Bicriteria Formulations: Properties

We briefly discuss the generality and the robustness of our bicriteria formulations. The discussion is based on the results in Marathe *et al.* [1995] and hence we keep the discussion brief.

We say that our formulation is robust since the quality of approximation is independent of which of the two criteria we impose the budget on. Specifically, the problem of finding a spanning tree that service all the nodes in a graph can be formulated in two natural ways : (i)(**Uniform service cost, Total cost, Spanning Tree**)-problem, and (ii)(**Total cost, Uniform service cost, Tree**)-problem. Problem (i) has already been discussed. In problem (ii), given a bound B on the cost of a spanning tree, we wish to find a spanning tree of cost no more than B such that the maximum service distance for any node not in the tree is minimized.

Note that these problems are meaningful only when the two criteria are *hostile* with respect to each other - the minimization of one criterion conflicts with the minimization of the other. A good example of hostile objectives are the degree and the total edge cost of a spanning tree in an unweighted graph. An example of a pair of objectives that are not hostile are the bottleneck cost (maximum cost of any edge) and the total edge cost of a spanning tree, since minimizing the latter automatically minimizes the former. Two minimization criteria are formally defined to be hostile whenever the minimum value of one objective is monotonically non-decreasing as the budget (bound) on the value of the other objective is decreased. It is easy to see that service cost and the total cost of the tree are hostile functions. Thus using the ideas in Marathe *et al.* [1995] we have the following result.

THEOREM 1. *Any* (α, β)*-approximation algorithm for (***Uniform service cost, Total cost, Spanning Tree***) can be transformed in polynomial time into a* (β, α)*-approximation algorithm for (***Total cost, Uniform service cost, Spanning Tree***).*

Theorem 1 directly extends to other variants of the problem such as the problems (**Uniform service cost, Diameter, Spanning Tree**), (**Uniform service cost, Total Cost, Steiner Tree**), etc. The extensions are immediate and thus we omit their proofs.

Next, we discuss the generality of our results. We claim that our results are more general because they subsume the case where one wishes to minimize some functional combination of the two criteria. For the purposes of illustration let **A** and **B** be two objective functions and let us say that we wish to minimize the sum of the two objectives **A** and **B**. Call this an (**A** + **B**, **S**) problem. The following theorem follows by arguments similar to those given in Marathe *et al.* [1995].

THEOREM 2. *Let* **ALG** *be any* (α, β)*-approximation algorithm for (***A***,* **B***,* **S***) on graph* G*. Then there is a polynomial time approximation algorithm* **ONE-ALG** *for the (***A** + **B***,* **S***) problem with performance guarantee* $(1 + \epsilon)\ \max\{\alpha, \beta\}$.

Similar results hold for the (**AB**, **S**) problem. In contrast, it is not clear how to extend an algorithm for the (**AB**, **S**) or the (**A** + **B**, **S**) problem to an (approximation) algorithm for the (**A**, **B**, **S**) problem. It is in this sense that we claim the generality of our results. Note that in some cases algorithms with better performance than **ONE-ALG** can be obtained directly for the unicriteria version of the problems (see Marathe *et al.* [1995]).

6. Hardness results

6.1 Different Costs

First we show the following hardness result for spanning trees under different cost functions, in the uniform service distance case, when all the service distances are the same. We use the recent results on the non-approximability of **MIN SET COVER** problem.

As an instance of the **MIN SET COVER** problem we are given a universe $Q = \{q_1, q_2, \ldots, q_n\}$ and a collection $Q_1, Q_2, \ldots, Q_m$ of subsets of Q. The problem is to find a minimum size collection of the subsets whose union is Q. Recently Feige [1996] has shown the following non-approximability result:

THEOREM 3. *Unless* **NP** $\subseteq$ **DTIME**$(n^{\log\log n})$*, the* **MIN SET COVER** *problem, with a universe of size k, cannot be approximated to better than a* $\ln k$ *factor.*

THEOREM 4. *Unless* **NP** $\subseteq$ **DTIME**$(n^{\log\log n})$*, the (***Uniform service cost, Total cost, Spanning Tree***) problem, with different cost functions, cannot be approximated to within (α, β), for any $\alpha \geq 1$ and any $\beta < \ln n$.*

PROOF. We show that for any $\alpha \geq 1$, if there is a polynomial-time (α, β)-approximation algorithm for the (**Uniform service cost, Total cost, Spanning Tree**) problem, then there is a polynomial-time β-approximation algorithm for the **MIN SET COVER** problem.

Construct the natural bipartite graph, one partition for set nodes and the other for element nodes, with edges representing element inclusion in the sets. To this bipartite graph, we add an enforcer node with edges to all the set nodes and also a mate node attached to the enforcer. Now we complete this *skeleton-graph* by throwing in all the edges. We set the d-cost of an edge from the enforcer to a set node to be 1. We set the d-cost of all other edges to be $\beta \cdot m + 1$. We now specify the c-costs (service costs) for the edges. We set the c-cost for the edge between the enforcer and the mate and for each edge between a set node and the element nodes contained in this set to be some fixed value, say S. We set the c-cost of all the edges between the enforcer and the set nodes to be $\alpha \cdot S + 1$. Let G denote the resulting instance (See Fig. 6.1) of the (**Uniform service cost, Total cost, Spanning Tree**) problem with the c and d cost functions as specified above and a uniform service budget of S.

It is easy to see that any collection of k subsets which form a set cover correspond to a tree in G that strictly services all the nodes and has a d-cost of k. This is because the tree consisting of the enforcer and the nodes corresponding to the sets in the collection, strictly services all the nodes and has a d-cost of k.

Let OPT denote the size of a minimum set cover to the original instance. Now we show that if there exists a tree T which is an (α, β)-approximation to the resulting instance G of the (**Uniform service cost, Total cost, Spanning Tree**) problem, then from it we can derive a β-approximation to the original set cover instance. Such a tree T must satisfy the following properties:

(1) The c-cost of T is at most $\beta \cdot$OPT. This follows from the definition of β-approximation and the fact that there exists a tree in G corresponding to OPT with d-cost at most OPT.

(2) The nodes of G must be serviced by T within budget S. This is because the c-cost of any edge is either S or $\alpha S + 1$, but T violates the budget constraint by at most a factor α.

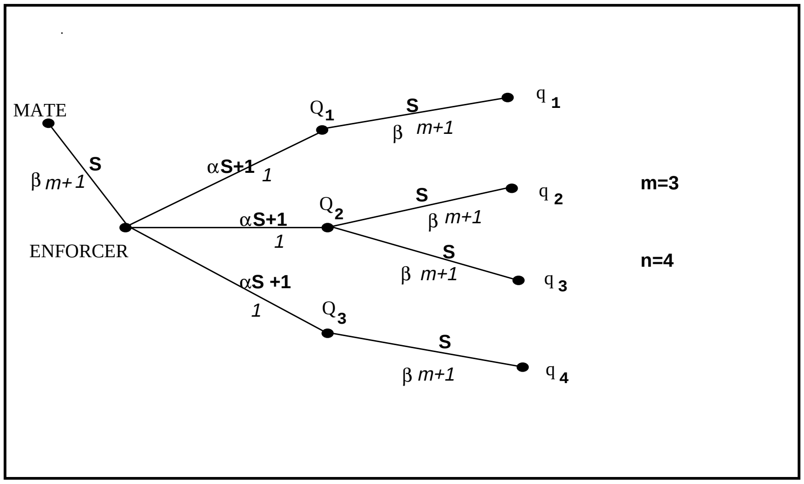

Fig. 6.1: The skeleton-graph depicting the reduction from **MIN SET COVER** to the (**Uniform service cost, Total cost, Spanning Tree**) problem. The c-costs indicated above the edges and the d-costs indicated below.

(3) The mate node cannot be in T. This is because the d-cost of any edge from the mate node is $\beta \cdot m + 1$ which is greater than the d-cost of T. Since only the enforcer node can service the mate node with a service cost of at most αS, the enforcer must be in T.

(4) Using the same reasoning as that for the mate node, none of the nodes representing the ground elements can be in T. To service these nodes, some of the set nodes must be in T.

We thus conclude that T consists only of the enforcer node and some of the set nodes. Since the d-cost of T is at most $\beta \cdot \textsc{Opt}$, it follows that the number of set nodes in T is at most $\beta \cdot \textsc{Opt}$. Since the element nodes are serviced by the chosen set nodes with a service distance of at most αS, the corresponding sets must form a set cover. We thus have a β-approximation algorithm for set cover and this completes the proof.

6.2 Identical costs

We first recall additional definitions and results.

Given an undirected graph $G(V, E)$, the **CONNECTED DOMINATION** problem (the optimization version), is to find a a dominating set D of vertices of minimum size such that the subgraph induced on D is connected. The result in Feige [1996] combined in straightforward fashion with the reduction in Garey and Johnson [1979] yields the following non-approximability result:

THEOREM 5. *Unless* **NP** $\subseteq$ **DTIME**$(n^{\log\log n})$, *the* **CONNECTED**

DOMINATION *problem, on a graph with n vertices cannot be approximated to better than a* $\ln n$ *factor.*

THEOREM 6. *Unless* **NP** $\subseteq$ **DTIME**$(n^{\log\log n})$, *the (***Uniform service cost, Total cost, Spanning Tree***) problem, with identical cost functions, cannot be approximated to within* (α, β)*, for* $\alpha < 2$ *and any* $\beta < \ln n$.

PROOF. We present an approximation preserving reduction from **CONNECTED DOMINATION** to the **(Uniform service cost, Total cost, Spanning Tree)** problem. Specifically, we show that if there is a $(< 2, \beta)$-approximation algorithm for the (**Uniform service cost, Total cost, Spanning Tree**) problem then there is a polynomial-time β-approximation algorithm for the **CONNECTED DOMINATION** problem.

Corresponding to an instance $G = (V, E)$ of **CONNECTED DOMINATION**, we create a complete edge weighted graph $G' = (V', E')$ as follows: we set $V' = V$. We set the c-cost of each edge in E' to be the length of the shortest path in G, and the uniform service budget S to be 1.

We claim that there exists a connected dominating set of size at most k in G if and only if there exists a solution to the (Uniform service cost, Total cost, Tree)-bicriteria problem with cost at most $(k-1)$. For the only if part, note that any spanning tree for connected dominating set of size k is a tree of cost $(k-1)$ that services all the nodes. Conversely, suppose we have a tree of cost $(k-1)$ servicing all the nodes in G'. Then, the tree has no more than k nodes, and all other nodes are at a distance of less than two (and hence at most one) from some node in the tree. So, the vertices in the tree form a connected dominating set for G. This completes the proof. □

7. Different Cost Functions

In this section, we present an $(1, O(\tilde{\Delta} \cdot \ln n))$-approximation algorithm for the (**Non-uniform service cost, Total cost, Tree**) problem with different cost functions. We first recall a few basic definitions and preliminaries.

DEFINITION 1. *A node u is said to* **service** *a node v if u is within distance* $\mathcal{S}_v$ *of* v*. The service-degree of a node is the number of nodes it services. The service-degree of the graph is the maximum over all nodes of the service-degree of the node and is denoted by* $\tilde{\Delta}$.

Given a graph G with edge weights and node weights, we define the **ratio weight** of a simple cycle $\mathcal{C}$ in G to be

$$\frac{\sum_{e \in \mathcal{C}} wt_e}{\sum_{v \in \mathcal{C}} wt_v}.$$

Here wt_e denotes the weight of an edge and wt_v denotes the weight of a vertex. In other words, the ratio weight of a cycle is the ratio of the edge weight of the cycle to the node weight of the cycle. As mentioned in Blum *et al.* [1996] the following problem is NP-hard:

DEFINITION 2. *The* **MIN-RATIO-ROOTED-CYCLE (MRRC)** *Problem: Given a graph $G = (V, E)$ with edge and node weights, and a distinguished vertex $r \in V$ called the root, find a simple cycle in G that contains r and has minimum ratio weight.*

By a slight modification of the ideas in Blum *et al.* [1996], we get the following theorem.

THEOREM 7. *There is a polynomial-time approximation algorithm with performance guarantee $\rho = O(1)$ for the* **MRRC** *problem.*

Finally, we assume that the graph is complete and the edge cost functions – c and d – obey triangle inequality. The reason for this is as follows: consider the (complete) graph obtained on the same set of vertices by adding edges between every pair of vertices of c and d-costs equal to that of the shortest c and d-cost paths between the corresponding vertices in the original graph; then any solution on this new graph transforms to a solution of identical value in the original graph.

7.1 Basic Technique

Before presenting the details, we give the main idea behind our algorithm. To begin with, we may assume that a specific node r belongs to the optimal tree. By running over all possible r's and picking the best we find the required (approximate) tree. The algorithm runs in phases. Initially, the solution contains only the node r. At any stage only a subset of the nodes are serviced by the set of solution nodes. Each phase the algorithm finds a nearly optimal minimum ratio weight cycle that services some of the remaining unserviced nodes in the graph. The cycle is contracted to a single node and the algorithm moves to the next phase. Termination occurs when all the nodes in the graph are serviced. A logarithmic performance guarantee is obtained by assuring that the cycle added in each phase has low cost compared to the optimal solution.

7.2 The Algorithm

We first define a few additional terms used in describing our algorithm. At any point in the algorithm, for each vertex $v_k \in V$, let B_{v_k} denote the set of vertices that are within c-distance of at most $\mathcal{S}_{v_k}$ from v_k. It is easy to see that B_{v_k} can be computed in polynomial time.

We also need the concept of *contraction* of a set of nodes in the graph. This is the natural operation of replacing this set of nodes with a single new node, deleting edges with both ends in the set, and making the new node the endpoint of those edges with exactly one endpoint in the set. ALGORITHM DIFFERENT describes the method in detail.

ALGORITHM DIFFERENT:

Input: A graph $G = (V, E)$, edge cost functions c and d, service budget $\mathcal{S}_{v_k}$ for vertex v_k under the c-cost function.

(I) For $i = 1$ to n do

(1) Set $r = v_i$. Set $j = 0$. Set $G_0 = G - B_r \cup \{r\}$.

(2) While $G_j \neq r$.

(a) Set $j = j + 1$.

(b) For all $v \in G_j$ compute B_v.

(c) Compute C_j, a ρ-approximate solution to the **MRRC** problem on G_j where the edge weights are the d-edge-costs and the node weights are $|B_{v_k}|$ for $k \neq i$ and 0 for r.

(d) Modify G_j by contracting C_j into a supernode. Set r to be the new supernode.

(e) Set $G_j = G_j - B_r \cup \{r\}$.

(3) Let $\mathcal{T}_i$ be a minimum spanning tree on $\bigcup_j C_j$ under the d-cost.

(II) Let HEU $= \min_i \mathcal{T}_i$. Output HEU.

Output: A tree HEU such that every vertex $v_i \in V$ is within a distance $\mathcal{S}_{v_i}$ from some node in HEU, under the c-cost, and the d-cost of HEU is at most $O(\tilde{\Delta} \cdot \ln n)$ times that of an optimal service-constrained tree.

7.3 Performance Guarantee

It is easy to see that ALGORITHM DIFFERENT outputs a tree that services all the nodes. It remains to show that the d-cost of HEU is within a factor of $O(\tilde{\Delta} \cdot \ln n)$ of the optimal. We prove this in the following two lemmas. For the rest of the paper, we will use the same symbol to denote a set and its cardinality and the intent will be clear from the context.

Let OPT denote an optimal tree. In what follows, let i denote that iteration of Step (I) in which $r = v_i \in$ OPT. Let f denote the number of iterations of Step 2 for this particular value of i. Let the set of cycles chosen in Step 2c of the algorithm be $C_1, \ldots, C_f$, in order. We use C_j to denote both the cycle as well as the d-cost of the cycle. We also use OPT and HEU to denote the d-costs of the corresponding tree. Let ϕ_j denote the number of nodes in G that are not serviced by the supernode r after choosing the cycle C_j in the jth iteration of Step 2. Alternatively, ϕ_j is the number of vertices in $G - r$ after Step 2e in the j'th iteration.. Thus, $\phi_0 \leq n$ while $\phi_f = 0$. Let cycle C_j service t_j new nodes.

LEMMA 1.

$$\frac{C_j}{t_j} \leq \frac{2\rho\tilde{\Delta}\mathrm{OPT}}{\phi_{j-1}}.$$

PROOF. Focus on the graph G_j at the end of iteration $j-1$. Since OPT services all the vertices we have that $\sum_{v_k \in \text{OPT}} |B_{v_k}| = |V(G_j)| = \phi_{j-1}$.

We first observe that OPT induces a cycle (by doing an Euler walk along the outside of the OPT tree, see Cormen *et al.* [1990], pp 697–700) with a ratio weight of $\frac{2\text{OPT}}{\phi_{j-1}}$. Hence, since in Step 2c we choose a ρ-approximate minimum ratio cycle C_j it follows that

$$\frac{C_j}{\sum_{v_k \in C_j} |B_{v_k}|} \leq \frac{2\rho\text{OPT}}{\phi_{j-1}}.$$

Since the service-degree of each vertex in G is at most $\tilde{\Delta}$, it follows that no vertex contributes more than $\tilde{\Delta}$ to the denominator of the left hand side in the above equation. Thus $\tilde{\Delta} \cdot t_j \geq \sum_{v_k \in C_j} |B_{v_k}|$. Hence

$$\frac{C_j}{\tilde{\Delta} \cdot t_j} \leq \frac{2\rho\text{OPT}}{\phi_{j-1}}.$$

The lemma follows. □

LEMMA 2.

$$\text{HEU} \leq 2\rho\tilde{\Delta}\ H_n \text{OPT}$$

where $H_n = 1 + \frac{1}{2} + \ldots + \frac{1}{n}$ *is the harmonic function.*

PROOF. By definition of ϕ_j and t_j, we have that

$$\phi_j = \phi_{j-1} - t_j \tag{7.1}$$

and from Lemma 1, we have

$$t_j \geq \frac{C_j \phi_{j-1}}{2\rho\tilde{\Delta}\text{OPT}} \tag{7.2}$$

Substituting Equation (7.2) into (7.1) we get

$$C_j \leq (2\rho\tilde{\Delta}\text{OPT})\frac{t_j}{\phi_{j-1}} \leq (2\rho\tilde{\Delta}\text{OPT})(H_{\phi_{j-1}} - H_{\phi_j}).$$

Hence, since $\phi_0 \leq n$ and $\phi_f = 0$, we get

$$\sum_{j=1}^{f} C_j \leq (2\rho\tilde{\Delta}\text{OPT})(H_{\phi_0} - H_{\phi_f}) \leq (2\rho\tilde{\Delta}\text{OPT})H_n.$$

The proof of the lemma now follows by observing that HEU$\leq \sum_{j=1}^{f} C_j$. □

Since $H_n \approx \ln n$ we obtain the following result.

THEOREM 8. *There is a* $(1, O(\tilde{\Delta} \cdot \ln n))$*-approximation algorithm for the (***Non-uniform service cost, Total cost, Tree***)-bicriteria problem with different cost functions, where* $\tilde{\Delta}$ *is the maximum service-degree of any node in the graph.*

Remark. Note that the bounds of Theorem 8 also extend to the Steiner version where only a set of terminal sites need to be serviced. The Steiner version reduces to the regular version by setting the service budgets of the nonterminal nodes to some large value, such as the diameter of the graph.

8. Identical Cost Functions

8.1 Spanning trees

We first consider the (**Non-uniform service cost, Total cost, Tree** problem for identical cost functions case and provide a $(2(1+\epsilon), 2(1+\frac{1}{\epsilon}))$-approximation algorithm. ALGORITHM IDENTICAL details this algorithm.

ALGORITHM IDENTICAL:

- *Input:* An undirected graph $G = (V, E)$, edge cost function c, service radius $\mathcal{S}_{v_k}$ for vertex v_k, an accuracy parameter $\epsilon > 0$.
- (1) For each node $v_k \in V$, let B_{v_k} denote the set of vertices that are within distance of at most $(1+\epsilon)\mathcal{S}_{v_k}$ from v_k.
  (2) Set $\mathcal{X}' = \{v_1, v_2, \ldots, v_n\}$. Set $\mathcal{X} = \emptyset$.
  (3) Repeat until $\mathcal{X}' = \emptyset$.
    (a) Let i be such that $\mathcal{S}_{v_i}$ is the least among all $v_i \in \mathcal{X}'$.
    (b) Set $\mathcal{X} = \mathcal{X} \cup \{v_i\}$.
    (c) Set $\mathcal{X}' = \mathcal{X}' \setminus \{v_k | B_{v_k} \cap B_{v_i} \neq \emptyset\}$.
  (4) Construct a graph G' on the set of vertices in $\mathcal{X}$. Let the cost of an edge in this graph be the distance of the shortest path between the two vertices in G.
  (5) Construct a minimum spanning tree T of G'.
  (6) Construct the subgraph H corresponding to T formed by replacing each edge in T by a shortest path of G.
  (7) Let HEU be a minimum spanning tree of H. Output HEU.
- *Output:* A tree HEU such that any vertex v_k is within a distance of $2(1+\epsilon)\mathcal{S}_{v_k}$ from some node in HEU and the cost of HEU is at most $2(1+\frac{1}{\epsilon})$ times that of any tree that contains a node within distance $\mathcal{S}_{v_k}$ of any vertex v_k.

Let OPT be an optimal solution. As mentioned we also use OPT and HEU to denote the cost of the corresponding trees. We prove the performance guarantee of ALGORITHM IDENTICAL in the following lemmas. Let the vertices in $\mathcal{X}$ at the termination of ALGORITHM IDENTICAL be $v_1, v_2, \ldots, v_f$, i.e., $|\mathcal{X}| = f$.

LEMMA 3. *Every vertex v_k is within a distance $2(1+\epsilon)\mathcal{S}_{v_k}$ of some vertex in $\mathcal{X}$.*

PROOF. If $v_k \in \mathcal{X}$ then the lemma follows since HEU contains v_k. If $v_k \notin \mathcal{X}$ then $\exists v_i \in \mathcal{X}$, such that $B_{v_i} \cap B_{v_k} \neq \emptyset$ and $\mathcal{S}_{v_i} \leq \mathcal{S}_{v_k}$. In this situation, it is easy to see that v_k is within a distance $(1+\epsilon)\mathcal{S}_{v_i} + (1+\epsilon)\mathcal{S}_{v_k} \leq 2(1+\epsilon)\mathcal{S}_{v_k}$. This completes the proof. □

LEMMA 4. $\text{OPT} \geq \displaystyle\sum_{i=1}^{f} \epsilon \mathcal{S}_{v_i}$.

PROOF. By definition, OPT contains at least one node from each B_{v_i} for all $v_i \in \mathcal{X}$ that is within distance $\mathcal{S}_{v_i}$ from v_i. Since the B_{v_i} for all $v_i \in \mathcal{X}$ are disjoint, any tree connecting these nodes must cross the peripheral $\epsilon\mathcal{S}_{v_i}$ width and the lemma follows. □

LEMMA 5. $\text{HEU} \leq 2(\text{OPT} + \sum_{i=1}^{f} \mathcal{S}_{v_i})$.

PROOF. We can construct a tree , spanning all the $v_i \in \mathcal{X}$ as follows: for each $v_i \in \mathcal{X}$, join v_i to a vertex in OPT that is within distance $\mathcal{S}_{v_i}$ by a shortest path. The length of this path is no more than $\mathcal{S}_{v_i}$. Thus, the cost of , is at most $\text{OPT} + \sum_{i=1}^{f} \mathcal{S}_{v_i}$. Note that , is a Steiner tree that spans all the vertices in $\mathcal{X}$. Since HEU is a minimum spanning tree on these same vertices, computed using shortest path distances between them, standard analysis of the minimum spanning tree heuristic for Steiner trees, yields that the cost of HEU is at most twice the cost of , . The lemma follows. □

LEMMA 6. $\text{HEU} \leq 2(1+\frac{1}{\epsilon})\text{OPT}$.

PROOF. Follows from Lemmas 4 and 5. □

Lemmas 3 and 6 yield the following theorem.

THEOREM 9. *For any $\epsilon > 0$ there is a $(2(1+\epsilon), 2(1+\frac{1}{\epsilon}))$-approximation algorithm for the (***Non-uniform service cost, Total cost, Tree***)-bicriteria problem with identical cost functions.*

8.2 Generalized Steiner Trees

We now consider the (**Uniform service cost, Total cost, Generalized Steiner forest**) problem – a generalization of the (**Uniform service cost, Total cost, Spanning tree**) problem. The reason for studying this problem stems once again from practical applications related to optical communication and distributed data management. Specifically, if we had a number of sites with connectivity requirements only among some subsets of the sites, rather than all the sites, then we need the solution subgraph to be a forest and not necessarily a tree. This motivates the service-constrained generalized Steiner forest problem. The formal statement of the (**Uniform service cost, Total cost, Generalized Steiner forest**) problem is as follows: given an undirected graph $G = (V, E)$ with two different cost functions c (modeling the service cost) and d (modeling the construction cost) for each edge $e \in E$, a set of k site pairs (s_i, t_i), and a bound $\mathcal{S}$ (on the maximum service constraint), find a minimum d-cost forest $\mathcal{F}$ such that for each site pair (s_i, t_i) there exists a tree $T \in \mathcal{F}$ with the property that both s_i and t_i are within distance $\mathcal{S}$ of the tree. The above problem is a generalization of the (**Uniform service cost, Total cost, Spanning tree**) problem. To see this, note that given an instance of **(Uniform service cost, Total cost, Spanning tree)**, we can construct an instance of the **(Uniform service cost, Total cost, Generalized Steiner forest)** problem by specifying a set of $n-1$ site pairs – one for each pair of vertices of the form (r, v) for some fixed node r in V. Clearly this implies that F consists of a single tree and the vertices not in the tree are appropriately covered.

8.3 Description of the Algorithm

We first define a few additional concepts that will be used to describe the algorithm.

For each site $v_i \in G$, let B_i (referred to as the ball around v_i), be the set of vertices that are within distance of at most $(1+\epsilon)\mathcal{S}$ from v_i. We refer to v_i as the *center* of B_i. It is easy to construct the set B_i in polynomial time. Let $\mathcal{B} = \{B_1, \ldots, B_n\}$ Given a set $\mathcal{B}$ of balls, we can naturally define an associated intersection graph $I(V_1, E_1)$. The vertices in I are in one-to-one correspondence with the balls in $\mathcal{B}$. There is an edge between two vertices in V_1 if and only if the corresponding balls have a non-empty intersection. (equivalently, if their centers are within a distance of $2(1+\epsilon)S$ from each other). As a part of our algorithm, we need to find minimum cost generalized Steiner forests. This problem is **NP**-hard, and we thus use the 2-approximation algorithm of Agrawal *et al.* [1995]. We denote this algorithm by **AKR-GEN-STEINER** for the rest of the section.

Algorithm Generalized Steiner gives details of our heuristic for approximately solving the (**Uniform service cost, Total cost, Generalized Steiner forest**) problem.

ALGORITHM GENERALIZED-STEINER:

- *Input:* An undirected graph $G = (V, E)$, edge cost function c, service budget $\mathcal{S}$, a set of site pairs (s_i, t_i) $1 \leq i \leq p$, an accuracy parameter $\epsilon > 0$.
- (1) If the distance between a site pair is at most $4(1+\epsilon)\mathcal{S}$, check if there is a node that is within a distance of at most $2(1+\epsilon)\mathcal{S}$ from both of them. If so discard this site pair from further consideration.
  (2) For each site $v_i \in G$, compute B_i .
  (3) Construct the intersection graph $I(V_1, E_1)$ corresponding to the balls $\{B_1, \ldots, B_n\}$.
  (4) Find a maximal independent set D in $I(V, E)$. Let $|D| = k$.
  (5) For each site s_i such that $s_i \notin \cup_{B_i \in D} B_i$ assign it to some site s_j such that B_i intersects with B_j and B_j is in D.
  (6) Mark all the balls in D as **active**. For each ball $B_i \in D$, update its set of sites as those that are assigned to this ball or contained within it.
  (7) Construct an auxiliary graph $G_2(V_2, E_2)$ as follows: The vertices V_2 are in one-to-one correspondence with the centers of the balls in D. The cost of an edge between two vertices equals the shortest path distance between the centers of the corresponding balls. The set of site pairs $(p_1, q_1), \ldots (p_m, q_m)$ are given as follows. If ball B_t is associated with site s_i and ball B_w is associated with site t_i, then the vertices corresponding to the balls B_t and B_w are considered site-pairs.
  (8) Construct a minimum cost generalized steiner tree T in $G_2(V_2, E_2)$ using Algorithm **AKR-GEN-STEINER**.
  (9) Output the tree T as the solution of the algorithm.
- *Output:* A forest HEU such that for all i, s_i and t_i are within a distance of $2(1+\epsilon)\mathcal{S}$ from some tree $T \in$ HEU and the cost of HEU is at most $8 + \frac{6}{\epsilon}$ times that of any optimal service constrained generalized Steiner forest.

Remarks:

(1) In Step 1, if a pair is discarded, it implies that the service constraint is violated by a factor of at most $2(1+\epsilon)$. This is because there is a trivial tree made of a single node at distance at most $2(1+\epsilon)S$ from both the sites.

(2) In Step 5 no pair of site-mates are assigned to the same ball due to the pruning in Step 1.

8.4 Performance Guarantee

By the definition of **active** balls, each ball in D separates at least one site. Since the balls are mutually disjoint it follows that any optimal forest has to visit each ball to ensure that the individual sites have at least one neighbor within a distance S from it. Thus any optimal service constrained forest must contain at least one node from each B_i in D.

Next, observe that $\text{OPT} \geq k\epsilon S$. To see this, observe that the above discussion implies that the optimal tree strictly visits each of the balls in D. Any forest connecting these nodes must cross the annular width of ϵS for each ball in D (since these balls are non-intersecting). Moreover, since there are k balls in D it follows that the total peripheral distance covered is at least $k\epsilon S$.

LEMMA 7. *There exists a generalized Steiner forest of the sites chosen in D of cost no more than $3k(1+\epsilon)S + \text{OPT}$.*

PROOF. The idea is to use the optimal generalized Steiner forest for the original problem of cost OPT. However since the specification of site pairs in D is different from those in the original graph, we must add more links to this solution to make it feasible for the new site pairs.

In particular, we first consider the connected components of site pairs under the new specification leading to subsets of nodes in D that are serviced by the same tree in the solution. We then identify a spanning tree of site-pair demands between these nodes - for example, two nodes x and y may be a site pair since their balls respectively intersect the balls of sites s_i and t_i in the original problem, so the edge (x, y) is a new site-pair demand. To satisfy this demand using the original optimal solution, we add the connections from x to s_i to the tree servicing s_i, and similarly from y to t_i to the tree servicing t_i (the same as before) thus connecting x and y via the tree. We do this for every site-pair demand in a spanning tree of demands for every such subset in D.

The cost of the extra connections added is the sum of the degrees of all the nodes in D in these spanning trees times $3(1+\epsilon)S$ for each connection. Since $|D| = k$ and the sum of degrees in a forest is at most twice the number of nodes, the cost of the connections is at most $6(1+\epsilon)S$. By the above argument, adding these connections to the original optimal forest gives a feasible solution for this problem with the stated cost. □

LEMMA 8. $\text{HEU} \leq 6(1+\epsilon)kS + 2\text{OPT}$.

PROOF. By Lemma 7, there is a generalized Steiner forest of cost at most $\text{OPT} + 3k(1+\epsilon)S$ connecting all the sites in D appropriately. Since we use a 2-approximation algorithm (Agrawal *et al.* [1995]), we get the bound claimed in the lemma. □

Thus we have the following theorem.

THEOREM 10. *There is a* $(2(1+\epsilon), 8+\frac{6}{\epsilon})$*-approximation algorithm for the* **(Uniform service cost, Total cost, Generalized Steiner forest)** *problem with identical cost functions.*

9. Diameter and Bottleneck

In this section we sketch our polynomial time results on two other variants of the objective function.

THEOREM 11. *The problems* **(Non-uniform service cost, Bottleneck, Spanning Tree)** *and (***Non-uniform service cost, Diameter, Spanning Tree***) with different cost functions, are solvable exactly in polynomial-time.*

PROOF. We first consider the (Non-uniform service cost, Bottleneck, Spanning Tree) problem – given a graph with two cost functions on the edges, and a service budget for each node, find a tree such that the service budget (under one cost function) for each node is satisfied and the tree has minimum bottleneck cost under the other cost function (i.e., the cost of the maximum edge in the tree is minimum). This problem can be solved by first sorting the edges in increasing order of the d-costs and adding the edges in that order until one of the connected components in the resulting subgraph satisfies the service constraints for all the nodes. The details are straightforward and so are omitted.

Next consider the **(Non-uniform service cost, Diameter, Spanning Tree)** problem. Using the ideas in Camerini and Galbiati [1982] and Ravi *et al.* [1996], one can show that the the service-constrained minimum diameter tree problem can be solved in polynomial time. In this problem, we are given a graph $G(V, E)$ and a service radius $\mathcal{S}_{v_i}$ for each vertex v_i. We wish to find a tree with minimum diameter (under the d-costs) such that every vertex v_i is within distance $\mathcal{S}_{v_i}$ (under the c-cost) from some node in the tree.

We only sketch the main idea of the algorithm below. The algorithm uses the roof graph construction in Ravi *et al.* [1996]. Consider the case when the d-costs are integral and polynomially bounded in the size of the graph. Consider OPT – a minimum-diameter service-constrained tree. Let OPT have diameter D. Let x and y be the endpoints of a longest path (under d-cost) in the tree. The weight of this path, D, is the diameter of the tree. Consider the midpoint of this path between x and y. It either falls at a vertex or in an edge in which case we can subdivide the edge by adding a new vertex. First we guess the value of D (there are only a polynomial number of guesses). All the potential midpoints lie in half-integral points along edges of which there are only a polynomial number. From each candidate point we consider the set of nodes within distance $D/2$ and check whether they service all the vertices in the graph. We choose the least such distance and the correspondingly suitable point and output the breadth-first tree rooted at this point appropriately truncated.

When the edge weights are arbitrary, the number of candidate midpoints are too many to check in this fashion. However, we can use a graphical representation (called the roof curve in Ravi *et al.* [1996]) of the distance of any node from any point along a given edge to bound the search for candidate points. This gives us the required result in the diameter case. □

Theorem 11 can be extended easily to the Steiner tree variant. We thus have the following theorem.

THEOREM 12. *The problems* **(Non-uniform service cost, Bottleneck, Steiner Tree)** *and (***Non-uniform service cost, Diameter, Steiner Tree***) with different cost functions, are exactly solvable in polynomial-time.*

10. Concluding Remarks

In this paper we focused on the problem of service-constrained network design problems. We formulated a number of these problems and presented general approximation techniques along with nearly-tight hardness results. In the bicriteria framework, we investigated problems **(A, B, S)**, where **A** = **Maximum service cost**, **B** $\in$ { **Total cost, Diameter, Bottleneck cost** } and **S** $\in$ { **Spanning tree, Steiner tree, Generalized Steiner tree** }.

The class of problems in which **A** = **Total service cost** is also a natural problem to study. Variants of this problem have been studied by Milo and Wolfson [1988]. Note that if both the objectives **A** and **B** are measured using identical costs then the problem (**Total service cost + Total Cost, Tree**) is the ubiquitious minimum spanning tree problem — and thus is efficiently solvable. In contrast, the **(Total service cost, Total Cost, Tree)** problem using identical costs can be shown to be **NP**-complete.

Acknowledgements

Research of the first author was supported by the Department of Energy under Contract W-7405-ENG-36. The second author acknowledges support from a DIMACS postdoctoral fellowship awarded at Princeton University in the early stages of this work. The research of the third author was supported by DARPA contract N0014-92-J-1799 and NSF 92-12184 CCR. We thank Professor S.S. Ravi (SUNY-Albany) for his collaboration in the early stages of the paper and discussion on related problems. We also thank Sven Krumke (University of Würzburg) for his constructive comments.

References

AGRAWAL, A., KLEIN, P., AND RAVI, R. 1995. When trees collide: An approximation algorithm for the generalized Steiner problem on networks. *SIAM Journal on Computing 24*, 440–456.

ARKIN, E. M., FEKETE, S. P., MITCHELL, J. S. B., AND PIATKO, C. D. 1994. Optimal Covering Tour Problems. In *Proceedings of the 5th International Symposium on Algorithms and Computation*.

ARKIN, E. M. AND HASSIN, R. 1994. Approximation algorithms for the geometric covering salesman problem. *Discrete Applied Math 55*, 197–218.

AWERBUCH, B., BARTAL, Y., AND FIAT, A. 1993. Competitive Distributed File Allocation. In *Proceedings of the 25th Annual ACM Symposium on the Theory of Computation*, 164–173.

AWERBUCH, B., FIAT, A., AND RABANI, Y. 1992. Competitive Algorithms for Distributed Data Manangement. In *Proceedings of the 24th Annual ACM Symposium on the Theory of Computation*, 39–50.

BLUM, A., RAVI, R., AND VEMPALA, S. 1996. A constant-factor approximation algorithm for the k-MST problem. In *Proceedings of the 28th Annual ACM Symposium on the Theory of Computation*, 442–448.

CAMERINI, P. C. AND GALBIATI, G. 1982. The bounded path problem. *SIAM Journal of Algebraic and Discrete Methods 3*, 4, 474–484.

CORMEN, T. H., LEISERSON, C. E., AND RIVEST, R. 1990. *Introduction to Algorithms*. MIT Press, Boston.

CURRENT, J. T. AND SCHILLING, D. A. 1989. The covering salesman problem. *Transportation Science 23*, 208–213.

DOWDY, L. AND FOSTER, D. 1982. Comparative Models of the File Assignment Problems. *ACM Computing Surveys 14*, 19–50.

FEIGE, U. 1996. A threshold of $\ln n$ for approximating set cover. In *Proceedings of the 28th Annual ACM Symposium on the Theory of Computation*, 314–318.

GAREY, M. R. AND JOHNSON, D. S. 1979. *Computers and Intractability: A guide to the theory of NP-completeness*. W. H. Freeman, San Francisco.

GREEN, P. E. 1992. *Fiber-Optic Networks*. Prentice-Hall.

HERNANDES, M., VILLALOBOS, J., AND JOHNSON, W. 1993. Sequential Computer Algorithms for Printed Circuit Board Inspection. In *Proceedings of SPIE – International Society of Optical Engineering, Intelligent Robots and Computer Vision XII: Active Vision and 3D Methods*, Volume 2056, 438–449.

IWANO, K., RAGHAVAN, P., AND TAMAKI, H. 1994. The Traveling Cameraman Problem with Applications to Automatic Optical Inspection. In *Proceedings of the 5th International Symposium on Algorithms and Computation (ISAAC), LNCS 834*, 29–37.

KUMAR, A. AND SEGEV, A. 1993. Cost and Availability Tradeoffs in Replicated Data Concurrency Control. *ACM Transactions on database Systems 18*, 102–131.

LUND, C., REINGOLD, N., WESTBROOK, J., AND YAN, D. 1994. On-Line Distributed Data Management. In *Proceedings of the 28th Annual ACM Symposium on the Theory of Computation*, 202–214.

MARATHE, M. V., RAVI, R., SUNDARAM, R., RAVI, S. S., ROSENKRANTZ, D. J., AND B., HUNT H. 1995. Bicriteria network design problems. In *Proceedings of the International Colloquium on Automata, Languages and Programming, LNCS 944*, 487–498.

MATA, C. AND MITCHELL, J. S. B. 1995. Approximation Algotrithms for Geometric Tour and Network Design Problems. In *Proceedings ACM Conference on Computational Geometry*.

MILO, A. AND WOLFSON, O. 1988. Placement of Replicated Items in Distributed Databases. *Proceedings of International Conference on Extending Database Technology (EDBT '88), LNCS 303*, 414–427.

RAVI, R., SUNDARAM, R., MARATHE, M. V., ROSENKRANTZ, D. J., AND RAVI, S. S. 1996. Spanning trees – short or small. *SIAM Journal on Discrete Mathematics 9*, 2 (May), 178–200.

WOLFSON, O. AND MILO, A. 1991. The multicast policy and its relationship to replicated data placement. *ACM Transactions on Database Systems 16*, 181–205.